1215419

THE UNITED STATES OF AMERICA

TO ALL TO WHOM THESE PRESENTS SHALL COME:

UNITED STATES DEPARTMENT OF COMMERCE

United States Patent and Trademark Office

November 20, 2004

THIS IS TO CERTIFY THAT ANNEXED HERETO IS A TRUE COPY FROM THE RECORDS OF THE UNITED STATES PATENT AND TRADEMARK OFFICE OF THOSE PAPERS OF THE BELOW IDENTIFIED PATENT APPLICATION THAT MET THE REQUIREMENTS TO BE GRANTED A FILING DATE.

APPLICATION NUMBER: *60/482,784*
FILING DATE: *June 25, 2003*
RELATED PCT APPLICATION NUMBER: *PCT/US04/20562*

Certified by

Jon W Dudas

Acting Under Secretary of Commerce
for Intellectual Property
and Acting Director of the U.S.
Patent and Trademark Office

CLAIMS

What is claimed is:

1. A method for allowing a user to connect to a privacy network comprising:

connecting to the network;

receiving a user name and password from a user; and

determining whether the user account is valid.

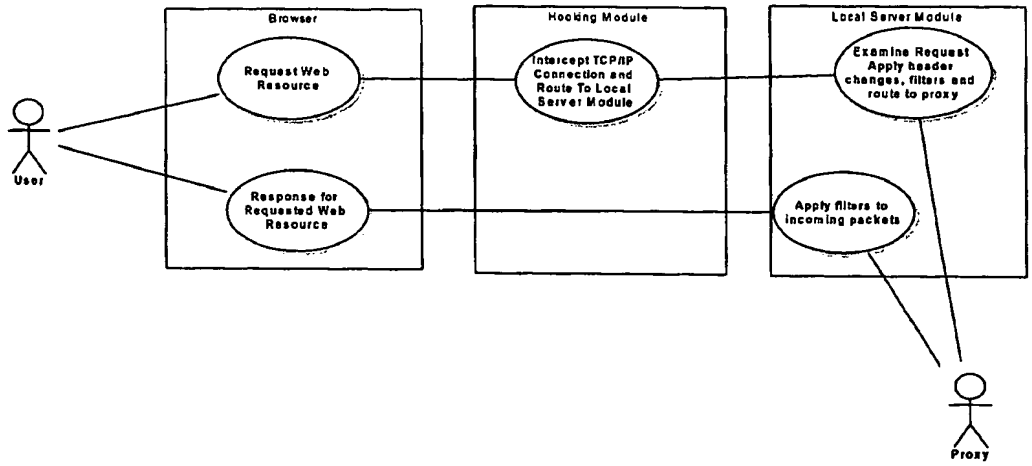
Browser
Hooking Module
Local Server Module
Request Web Resource
Intercept TCP/IP Connection and Route To Local Server Module
Examine Request Apply header changes, filters and route to proxy
Response for Requested Web Resource
Apply filters to incoming packets
User
Proxy

FIG. 1

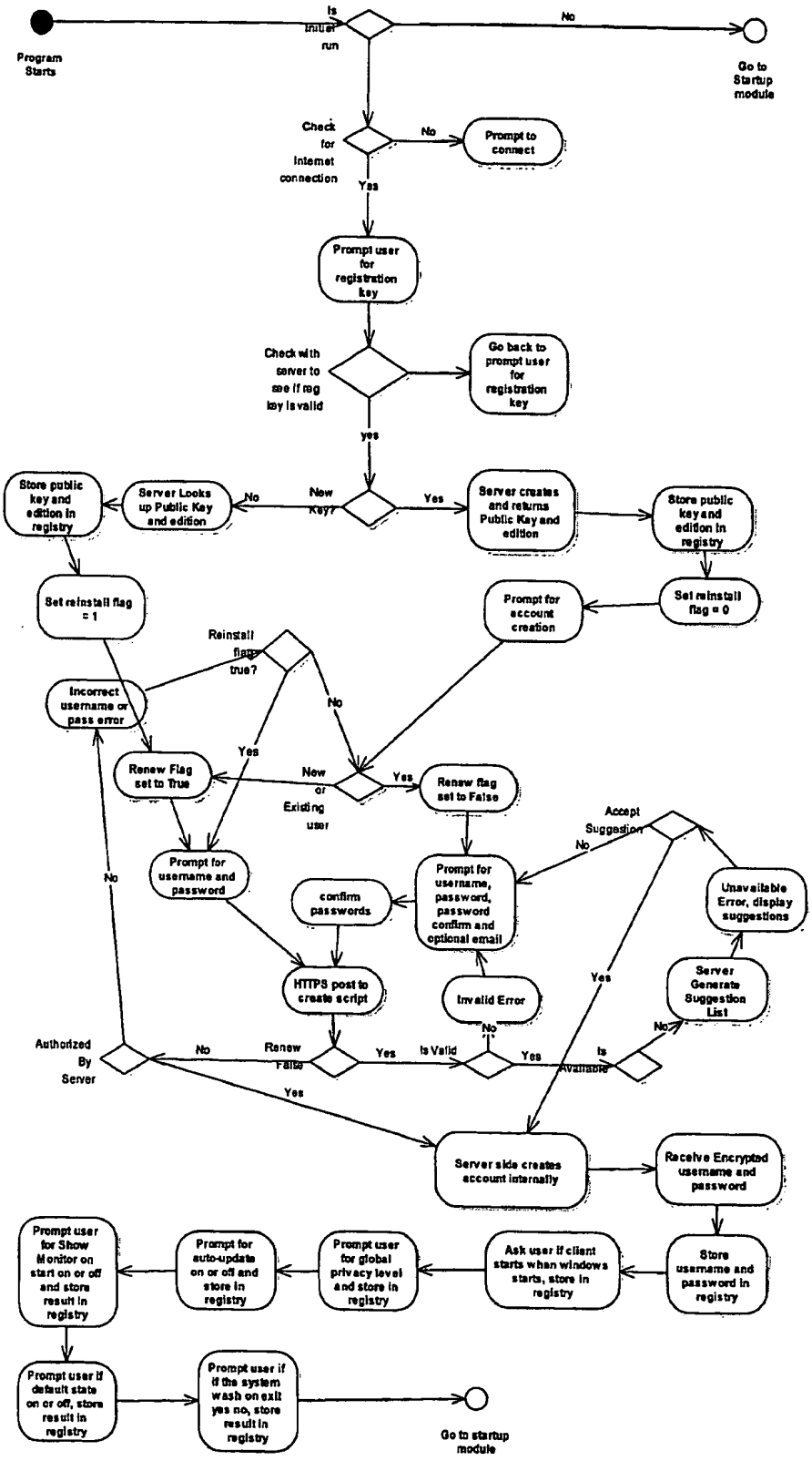
Program Starts
Is Initial run
No
Go to Startup module
Check for Internet connection
No
Prompt to connect
Yes
Prompt user for registration key
Check with server to see if reg key is valid
Go back to prompt user for registration key
yes
Store public key and edition in registry
Server Looks up Public Key and edition
No
New Key?
Yes
Server creates and returns Public Key and edition
Store public key and edition in registry
Set reinstall flag = 1
Set reinstall flag = 0
Prompt for account creation
Reinstall flag = true?
Incorrect username or pass error
No
Yes
Renew Flag set to True
New or Existing user
Yes
Renew flag set to False
Accept Suggestion
No
Prompt for username and password
No
confirm passwords
Prompt for username, password, password confirm and optional email
Unavailable Error, display suggestions
HTTPS post to create script
Invalid Error
No
Yes
Server Generate Suggestion List
No
Authorized By Server
No
Renew = False
Yes
Is Valid
Yes
Is Available
Yes
Server side creates account internally
Receive Encrypted username and password
Prompt user for Show Monitor on start on or off and store result in registry
Prompt for auto-update on or off and store in registry
Prompt user for global privacy level and store in registry
Ask user if client starts when windows starts, store in registry
Store username and password in registry
Prompt user if default state on or off, store result in registry
Prompt user if if the system wash on exit yes no, store result in registry
Go to startup module

FIG. 2

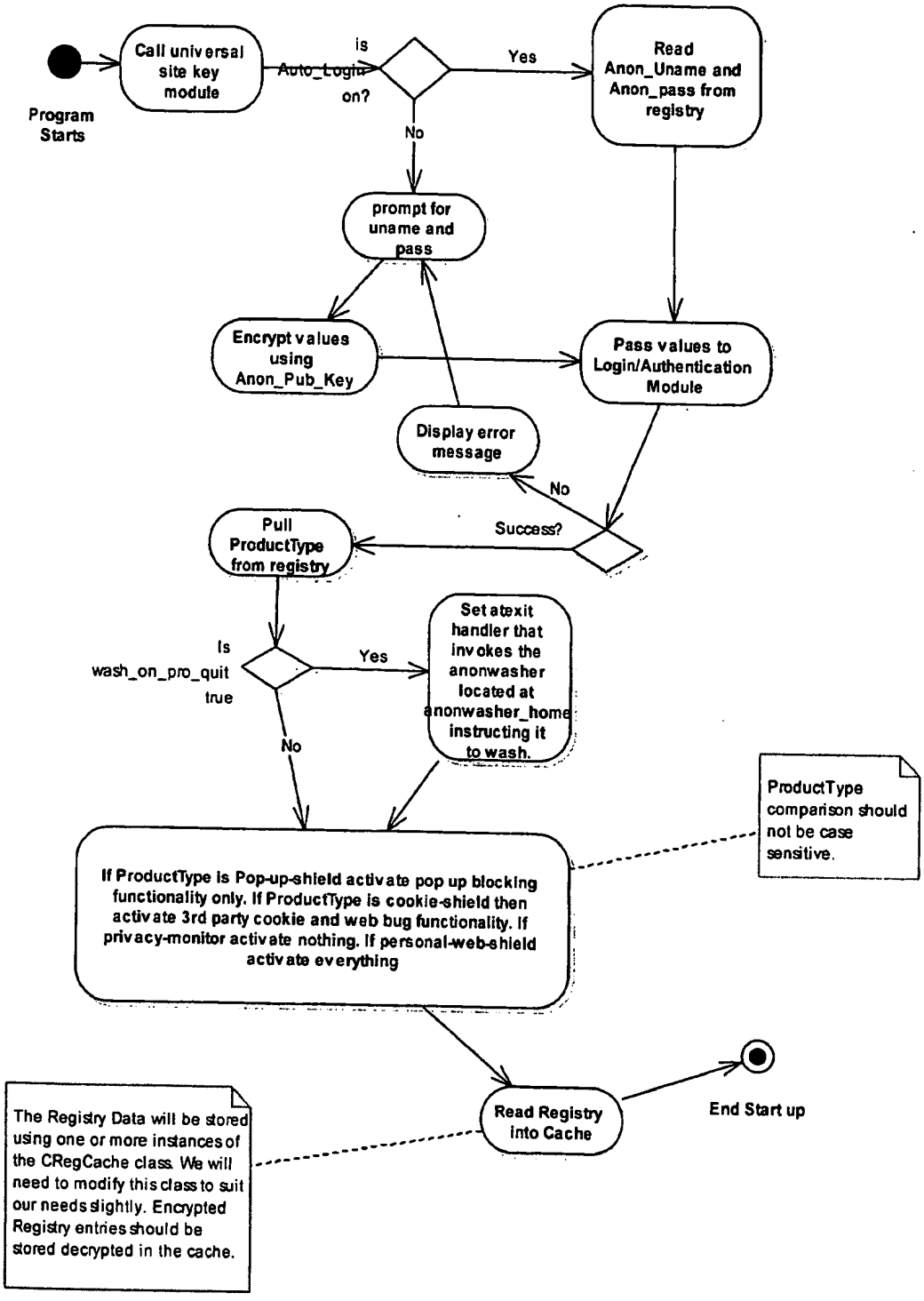
Program Starts
Call universal site key module
is Auto_Login on?
Yes
No
Read Anon_Uname and Anon_pass from registry
prompt for uname and pass
Encrypt values using Anon_Pub_Key
Pass values to Login/Authentication Module
Display error message
No
Success?
Pull ProductType from registry
Is wash_on_pro_quit true
Yes
No
Set atexit handler that invokes the anonwasher located at anonwasher_home instructing it to wash.
ProductType comparison should not be case sensitive.
If ProductType is Pop-up-shield activate pop up blocking functionality only. If ProductType is cookie-shield then activate 3rd party cookie and web bug functionality. If privacy-monitor activate nothing. If personal-web-shield activate everything
Read Registry into Cache
End Start up
The Registry Data will be stored using one or more instances of the CRegCache class. We will need to modify this class to suit our needs slightly. Encrypted Registry entries should be stored decrypted in the cache.

FIG. 3

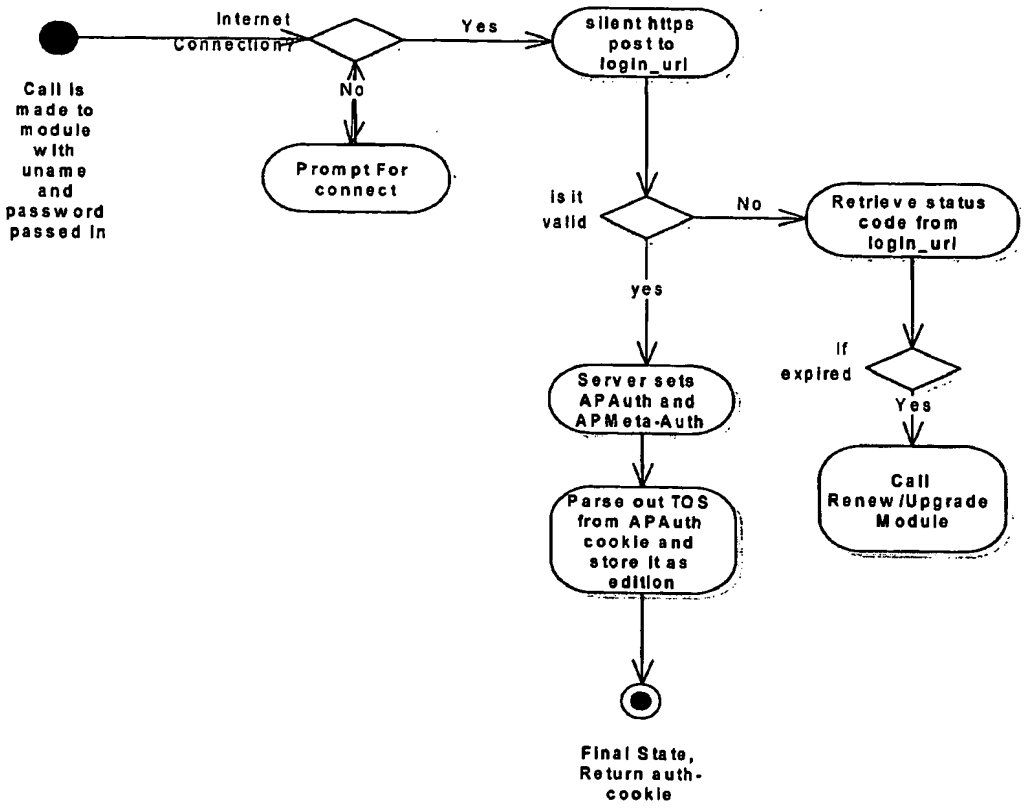
Call is made to module with uname and password passed in
Internet Connection?
Yes
No
Prompt For connect
silent https post to login_url
is it valid
No
yes
Retrieve status code from login_url
If expired
Yes
Call Renew/Upgrade Module
Server sets APAuth and APMeta-Auth
Parse out TOS from APAuth cookie and store it as edition
Final State, Return auth-cookie

FIG. 4

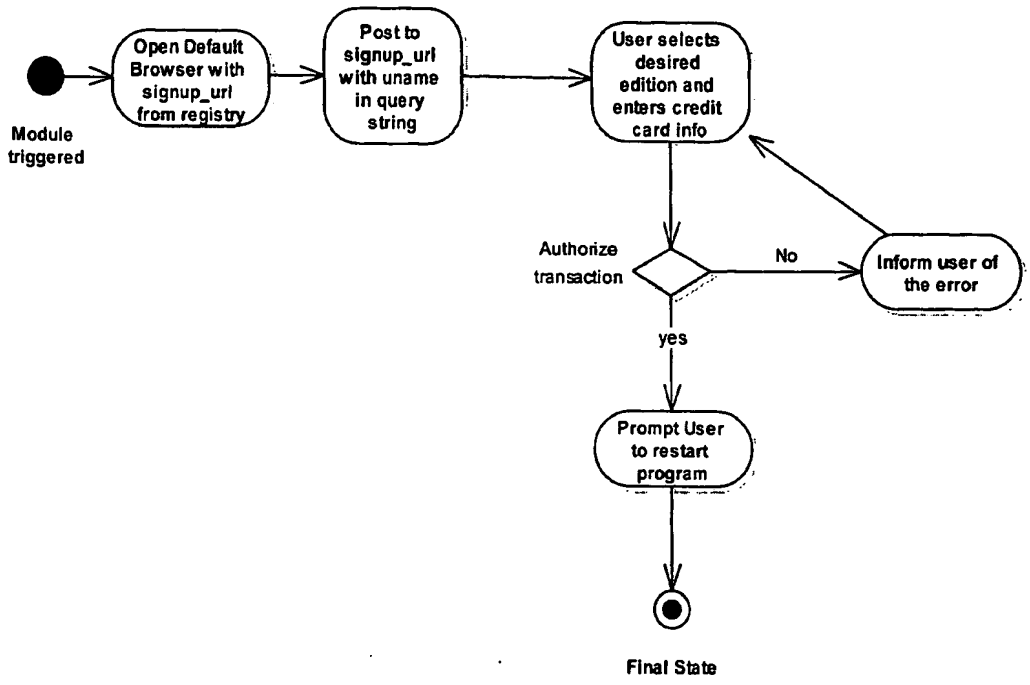
Module triggered
Open Default Browser with signup_url from registry
Post to signup_url with uname in query string
User selects desired edition and enters credit card info
Authorize transaction
No
Inform user of the error
yes
Prompt User to restart program
Final State

FIG. 5

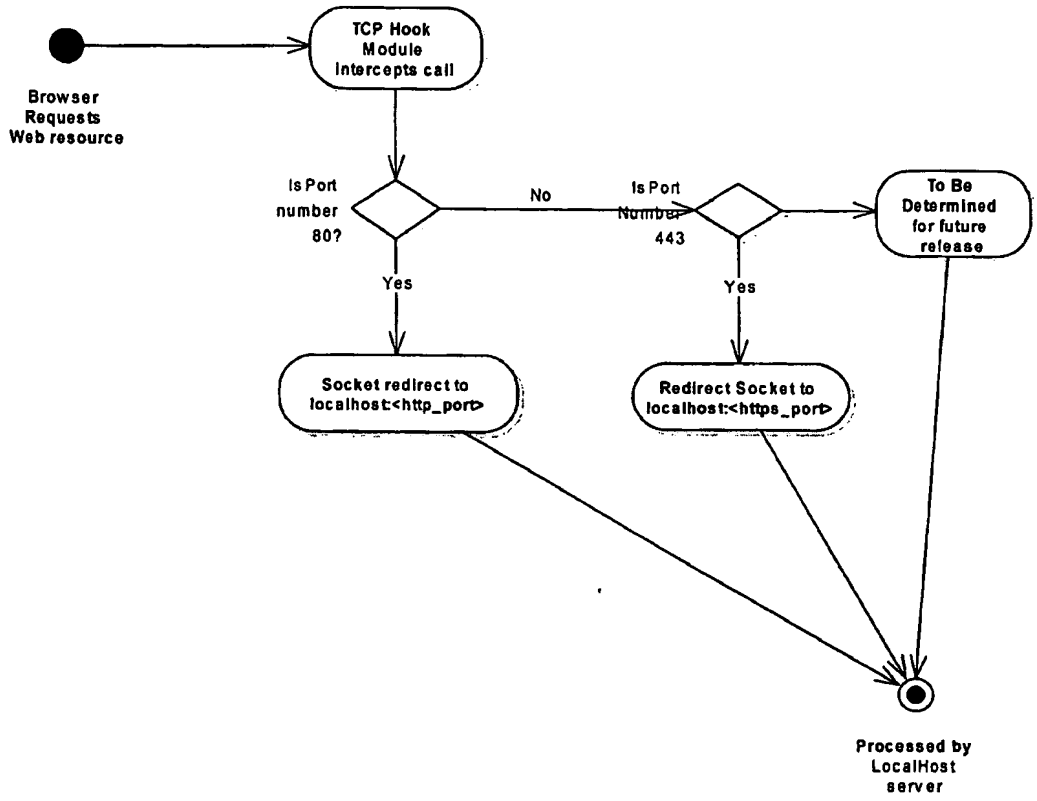
Browser
Requests
Web resource
TCP Hook
Module
Intercepts call
Is Port
number
80?
No
Is Port
Number
443
To Be
Determined
for future
release
Yes
Yes
Socket redirect to
localhost:<http_port>
Redirect Socket to
localhost:<https_port>
Processed by
LocalHost
server

FIG. 6

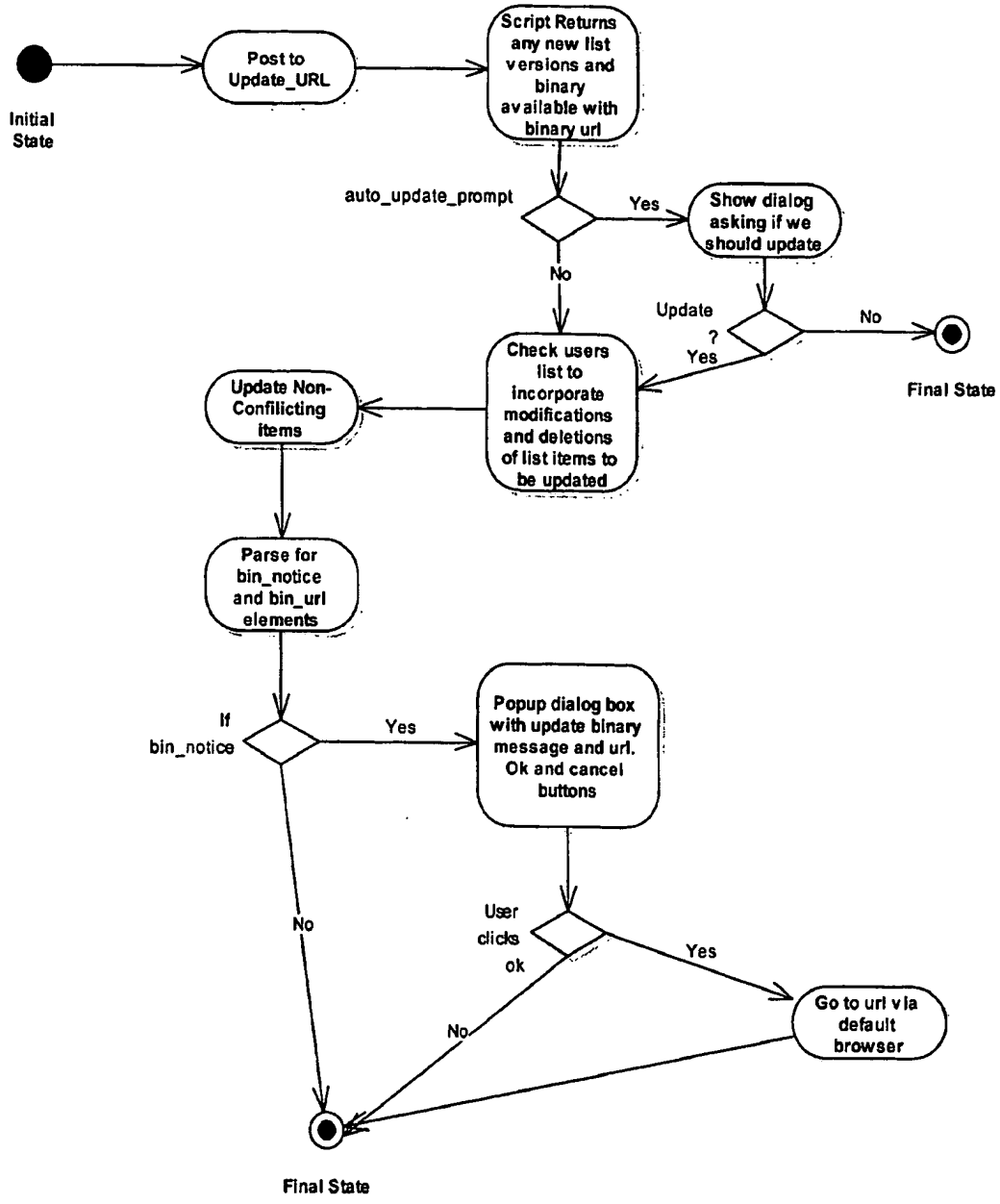
Initial State
Post to Update_URL
Script Returns any new list versions and binary available with binary url
auto_update_prompt
Yes
Show dialog asking if we should update
No
Update ?
No
Yes
Final State
Check users list to incorporate modifications and deletions of list items to be updated
Update Non-Confilicting items
Parse for bin_notice and bin_url elements
If bin_notice
Yes
Popup dialog box with update binary message and url. Ok and cancel buttons
No
User clicks ok
Yes
No
Go to url via default browser
Final State

FIG. 7

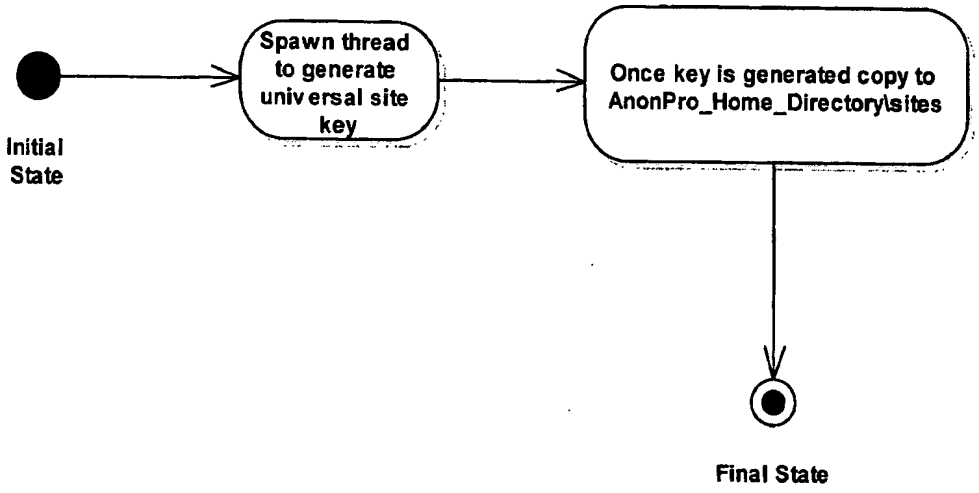
Spawn thread to generate universal site key
Once key is generated copy to AnonPro_Home_Directory\sites
Initial State
Final State

FIG. 8

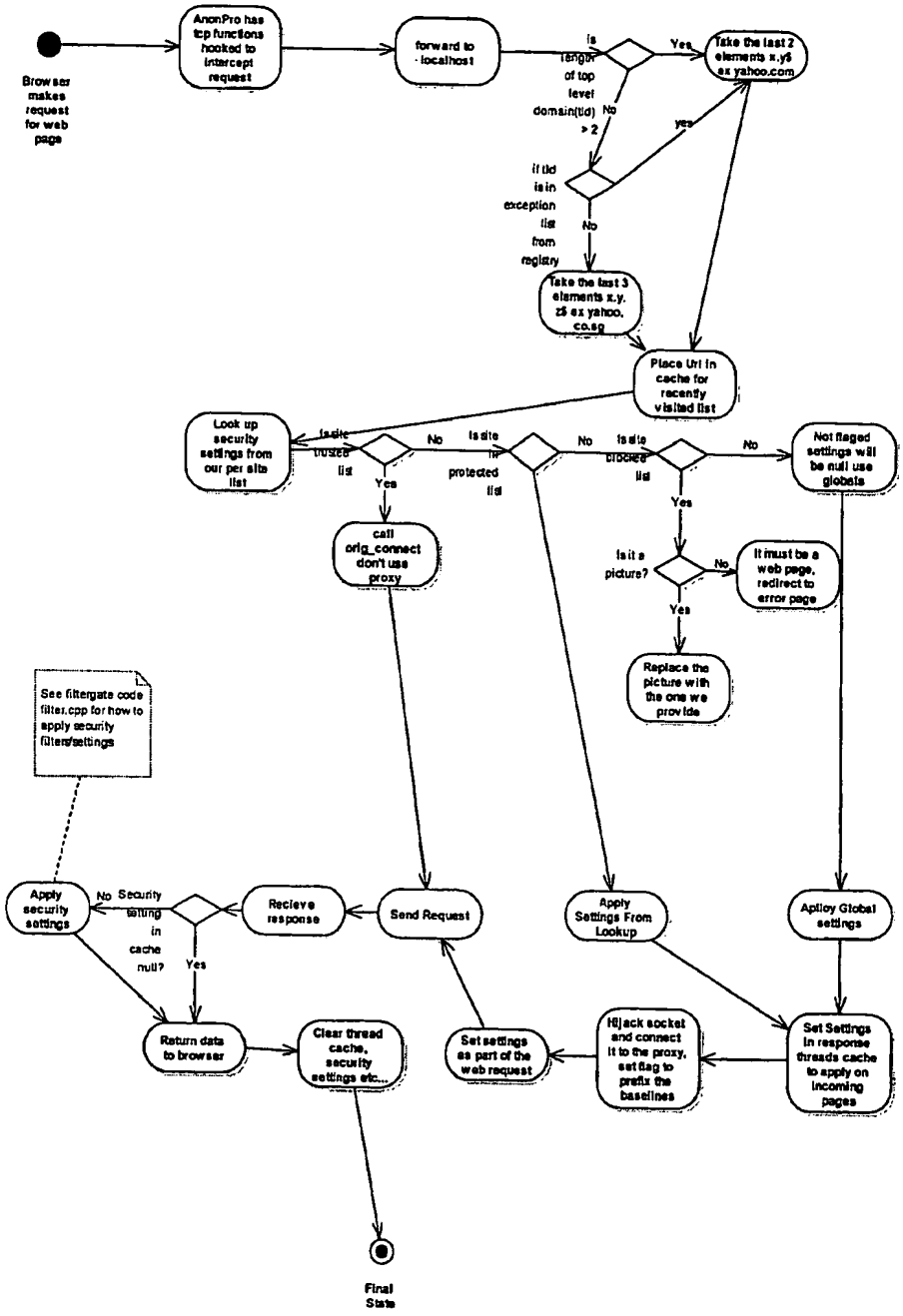
Browser makes request for web page
AnonPro has tcp functions hooked to intercept request
forward to localhost
Is length of top level domain(tld) > 2
No
Yes
Take the last 2 elements x.y$ ex yahoo.com
yes
if tld is in exception list from registry
No
Take the last 3 elements x.y. z$ ex yahoo. co.sg
Place Url in cache for recently visited list
Look up security settings from our per site list
Is site trusted list
No
Yes
Is site in protected list
No
Is site blocked list
No
Yes
Not flaged settings will be null use globals
call orig_connect don't use proxy
Is it a picture?
No
Yes
It must be a web page, redirect to error page
Replace the picture with the one we provide
See filtergate code filter.cpp for how to apply security filters/settings
Apply security settings
No
Security setting in cache null?
Yes
Recieve response
Send Request
Apply Settings From Lookup
Aplloy Global settings
Return data to browser
Clear thread cache, security settings etc...
Set settings as part of the web request
Hijack socket and connect it to the proxy, set flag to prefix the baselines
Set Settings in response threads cache to apply on incoming pages
Final State

FIG. 9

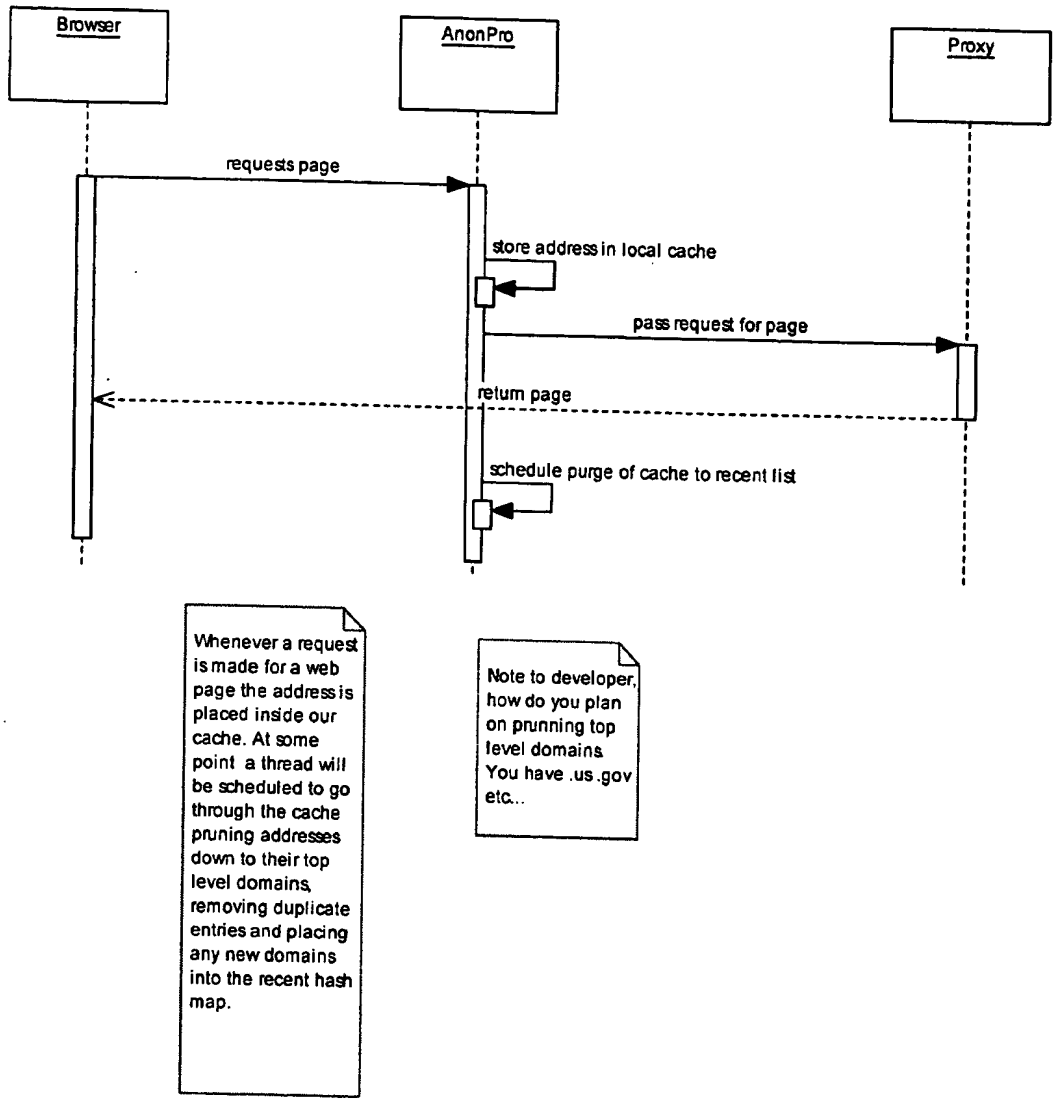
Browser
AnonPro
Proxy
requests page
store address in local cache
pass request for page
return page
schedule purge of cache to recent list
Whenever a request is made for a web page the address is placed inside our cache. At some point a thread will be scheduled to go through the cache pruning addresses down to their top level domains, removing duplicate entries and placing any new domains into the recent hash map.
Note to developer, how do you plan on prunning top level domains. You have .us .gov etc...

FIG. 10

U.S. Express Mail No.: EU245163300US
Filing Date: June 25, 2003

PATENT
Docket No. IF03004USV

A NETWORK PRIVACY SYSTEM

Inventors

LANCE M. COTTRELL
JAMES A REYNOLDS
DARYA MAZANDARANY
PELEUS UHLEY
&
GENE NELSON

BACKGROUND OF THE INVENTION

[001] **1. Field of the Invention.**

[002] This invention relates generally to network communication systems. In particular, this invention relates to an Internet privacy system capable of operating across multiple platforms.

[003] **2. Related Art.**

[004] As the global computer network known as the Internet continues to grow globally at a rapid pace, an increasing number of people and businesses from around the world are accessing the Internet for both business and personal activities. As a result, the Internet has become a virtual community were people communicate with each other by sending and receiving electronic, voice and image messages for business and pleasure. These communications include sharing ideas and information, sending personal and business message back and forth, researching information, expressing opinions and ideas both personal and political, and conducting business negotiations and transactions

(generally known as "electronic commerce" or "e-commerce"). In response to this new electronic activity, business, governments and certain individuals attempt to identify and track individual Internet users for numerous purposes including, but not limited to, advertising, market research, customizing information of Internet sites (i.e., "websites") snooping and ease dropping on communications, political and law enforcement activities, fraud and malicious activities. Many of these attempts are threats to the individual users of the Internet because they attempt to gain personal information about the user and the user's activities on the Internet (generally referred to as the user's "online activities") typically without the user's express consent or knowledge.

[005] These threats typically gain information about the user by logging a user's Internet Protocol ("IP") address (the electronic address that specifically identifies a user's computer to the network) or by installing programs or files on to the user's computer such as "cookies," ActiveX™ applications, Java™, script files, Spyware, or hostile programs such as viruses. These threats allow an outside user, be it a government, business, or individual entity, to perform such tasks as identify a user, obtaining the user's personal information that is stored on the computer (including names, address, financial, private files, and/or other confidential, private and/or sensitive information), and track the user's activities on the Internet including recording every website visited or every email sent or received by the user. Malicious programs such as viruses may also be installed on the user's computer that can modify, erase or destroy the user's operating system of personal files.

[006] Unfortunately, most people that utilize the Internet do not understand technically how networks such as the Internet function nor do they generally appreciate the number and types of threats that they will experience once they connect (i.e., "log-on") to the Internet. Past attempts at protecting users on the internet include using "firewalls" to block certain types of threats from the Internet, virus protection programs for detecting malicious programs, and spyware and cookie file removal software. However, these past attempts do not protect a user's identity because most of these approaches attempt to disinfect a user from intruders after the fact. These past approaches do not protect a user's identity as soon as the user connects to the Internet because connected websites are able to read and identify the user's IP address among other things. A need therefore exist to protect a user's identity as soon as the user connects to the Internet (i.e., known as "surfing the web" or "surfing the Net").

[007] Attempts in the past at protecting the user's identity have included allowing a user to connect to an intermediate server connected to the Internet that extracted off the user's IP information and substituted it with the IP address of the intermediate server thus creating an anonymous user that could then continue to surf the Net without worrying that their IP information would be used to identify them.

[008] Unfortunately, this approach was too technical and difficult to operate by most Internet users. Therefore, there is a need for a privacy management system that solves the problems recited above and allows Internet users to easily maintain their privacy by utilizing an anonymous server.

BRIEF DESCRIPTION OF THE FIGURES

[009] The components in the figures are not necessarily to scale, emphasis instead being placed upon illustrating the principles of the invention. In the figures, like reference numerals designate corresponding parts throughout the different views.

[010] FIG. 1 shows a block diagram an example implementation of the invention.

[011] FIG. 2 shows a flow chart of an example process performed by a Startup Wizard Module.

[012] FIG. 3 shows a flow chart of an example process performed by a Startup Module.

[013] FIG. 4 shows a flow-chart of an example process performed by a Login/Authentication module.

[014] FIG. 5 shows a flow-chart of an example process performed by a Renew /Upgrade Module.

[015] FIG. 6 shows a flow-chart of an example process performed by a TCP Hook module.

[016] FIG. 7 shows a flow-chart of an example process performed by a Client Update Module.

[017] FIG. 8 shows a flow-chart of an example process performed by a Universal site key module.

[018] FIG. 9 shows a flow-chart of an example process performed by a Non-SSL Module.

[019] FIG. 10 shows a signal flow diagram for an example process performed by the invention for recently visited sities.

DETAILED DESCRIPTION

[020] This invention describes a method for providing Internet privacy service which shall be described in relation to example implementation herein referred to as AnonPro. AnonPro may be a specific implementation of our inventions for providing Internet privacy services. The components described in this detailed description and figures are an example implementation for some of our particular applications, however, the technologies and inventions described herein are much more general. The components are generally a network level traffic interceptor (client side), a client proxy, a server proxy, an SSL module, and some web based services (such as the user authentication, server lists, recommended site settings lists etc.). Generically speaking, the combination of the components is an "Internet privacy system." The client part is "Internet Privacy Client", while the server proxy is "Internet Privacy remote proxy." In the description of this invention we often refer to registry entries or other specific ways of storing information. In all cases this information could be stored in any number of ways including in flat files, indexed files, local or remote databases, among others. In the description of this invention we often refer to cookies. Many other information transfer techniques could be used in place of cookies including HTML headers, changes to URLs or other addresses, any other standard or custom message or data structure. In the description of this invention we often refer to XML data structures. In general these

structures could be replaced with any other kind of data structure, including other standard and non-standard, encrypted and non-encrypted structures. CA stands for "Certificate Authority" and refers to an entity or encryption key used for signing other keys such as SSL keys.

[021] Additionally, The AnonPro Server Proxy (also known as the "Internet Privacy Remote Proxy") is a system that relays data from the client on the user's PC to the computer hosting the content or service the user is trying to access through the system's Internet Privacy System (the Destination). The proxy acts to hide the user's IP address and may perform other actions based on the content of the request or the contents of the reply from the Destination. These actions may include adding, changing, or removing text, data, information, scripts or other content from either the data from the user to the destination, or from the destination back to the user. The Internet Privacy Remote Proxy is not used in all modes of the Internet Privacy Client. In some modes the Internet Privacy Client connects directly to the Destination. Whether or not the Internet Privacy Remote Proxy is used depends on the privacy settings the user has set for that particular site. The Internet Privacy Remote Proxy is only used if the hiding of the user's IP, or the other changes the Remote Proxy makes to the data, are required for the particular settings. Otherwise the connection is direct.

[022] This invention builds on the features and functions of previous projects such as Anonymizer 2.0 (designed and operated by Anonymizer, Inc. of San Diego, California). Some components in the proxy server may need to be modified and at this point it is not known what the scope is.

[023] The main goal of this invention is to shield the user from the various threats presented when using the Internet as shown in FIG. 1. To achieve this we embed ourselves in the tcp layer of the system and route the client's data to a transparent proxy running on the client. The client proxy may determine what filters to apply based on the users settings and then forward the data to the anonymizer proxy to perform additional security measures. The server cert will be used to do silent SSL connections with the Web-Server to send usernames, passwords etc.

[024] In FIG. 2 a Startup Wizard Module is described. When the client is started, it is loaded into the system tray and displayed as an icon. Immediately after that, the client checks to see if the client is being run for the first time if so the client startup wizard (REFERENCE SCREEN SHOT) is invoked. The wizard will check for an Internet connection and prompt the user to open one if no such connection is found. Next it will prompt the user to enter his/her registration this will be posted to the registry value Account_Creation_URL with a query string like "?action=regval&regcode=<registration code>" this will return a document Account_Creation_URL return. A return value of 0 means the key is invalid and a message should be displayed followed by reprompt for key otherwise the product edition is to be stored in the registry (ProductType) and the public key for the user will be stored in the registry (Anon_Pub_Key) and the user will start the account creation process. The wizard will ask the user if he/she is new or existing if they are existing they will be prompted for their username and password if they are a new user they will be prompted for their desired username, password, password confirmation and optional email address. The wizard will verify that the two passwords match throwing an

error message if they don't then lookup the account creation URL (Account_Creation_URL) and do a https post to it passing in the user info as a query string if renew flag is 0 like querystring="?action=createacct&renewflag=<renewflag>&reinstallflag=<reinstallflag> &uname=<username>&passwd=<password>&email=<email>" where <renewflag> is either 0 for new user or 1 for existing and <reinstallflag> is 0 for unused registration key and 1 for a used registration key. This post will return an XML document Account_Creation_URL return. The wizard will then prompt the user if he/she would like the system to start when windows starts (or they login). We will also prompt for auto update on or off and write that into the Auto update registry entry. Next we will prompt to see if we should show the privacy monitor on startup. Now we need to prompt the user to see if the default state of the system is on or off and store the result in the Default_state registry entry. Finally we will prompt the user to see if we should run the other programs on exit or not.

[025] FIG.3 shows a flow-chart for a process performed by a startup module. The startup module handles the flow of the client when it starts. First it will call the universal site key module then it will check the Auto_login registry setting to see if it should prompt for the username and password. If Auto_Login is enabled the Anon_Uname and Anon_Password will be passed to the Login/Authentication Module. If Auto_Login is not enabled then we will use the Anon_Pub_Key to encrypt the username and password that the user entered and pass those values to the Login/Authentication Module. If the return

value of the module is not success then we may throw an error message and reprompt for the username and password.

[026] FIG. 4 shows a flow-chart of an example process performed by a Login/Authentication module. This module may be called whenever a client process (i.e Client Proxy or the SSL server, etc...) wants to login to the anonymizer service and receive an APAuth cookie to proceed. If auto_login is on the calling process should pass Anon_Uname and Anon_Password in as the parameters. If the auto_login is not set the calling process will prompt the user for the user name and password and then encrypt them using Anon_Pub_Key these will then be passed in as parameters. The module when called will do a silent HTTPS post to Login_URL with querystring like "?uname=<username> & passwd=<password>". The output of this post will be xml in this form:

[027] <root>

[028] <status>invalid||expired||inactive||active</status>

[029] </root>

[030] If successful APAuth and APMeta-Auth will be set and we should parse out the <tos> value from the APAuth cookie. This value will be stored in the ProductType registry entry. If not successful we will check to see if the user is expired if so call the update/renew module. If there is any other error it will be returned to the calling process, which will be responsible for re-prompting the user for his/her info.

[031] FIG. 5 shows a flow-chart of an example process performed by a Renew /Upgrade Module. This module will be called whenever we want to prompt the user to

upgrade or renew their product. When invoked it will open the default browser with the Signup_URL that we have stored in the registry posting the following query string like "?uname=<encrypted username>". From here the user will select the edition he/she wants to renew or upgrade and enter his/her info. Upon successful completion it will prompt the user to restart the client.

[032] FIG. 6 shows a flow-chart of an example process performed by a TCP Hook module. This module is responsible for hooking all the necessary tcp layer calls and redirecting the data to the client proxy module. When the browser tries to connect a socket the TCP Hook module will intercept the call and examine the port number. If it is 80 it will redirect the socket to localhost with the port set to the registry value (http_port). If it is port 443 it will redirect the socket to localhost with the port set to the registry value (https_port). If it is any other port it will redirect to localhost with port 8080.

[033] FIG. 7 shows a flow-chart of an example process performed by a Client Update Module. When the client update module is called it will make a silent http post to the Update_URL. This script will return any new list versions and a bin_notice with a bin_url based off the information in the AnonPro cookie. If Auto_update_prompt is set to 1 then we will pop up a dialog to see if we should update the lists. If they answer yes we proceed if they answer no we do no update. Before writing each list into the registry we will read TrustedList, ProtectedList, BlockedList and ServerList from the registry and check for modifications or deletions by the user. If the user has done either of these then their changes should persist. If we have parsed out a bin_notice and bin_url then we will pop up a dialog informing the user of the binary update and asking them if they would

like to download it. If they choose ok we will open the default browser and navigate to that url.

[034] A client Proxy Module is to be run possibly separate from the client. When the client proxy module starts up it will read the TrustedList, ProtectedList and the BlockedList from the registry in order to populate our per site preferences cache.

[035] FIG. 8 shows a flow-chart of an example process performed by a Universal site key module. This module will be invoked once on startup to generate a new universal site key to use for the current session. This key will be used to forge the identity of all ssl servers the client tries to connect to. It will be invoked as a low priority thread to run in the background and generate the key. Once the process is finished it will swap the new key for the current one and exit. To generate this key pair we may use the OpenSSL functions to create the RSA key pair. Once generated this key pair is it will be stored encrypted using the SHA-1 hash of the username and password in AnonPro_Home Directory\sites. We will also write a function to generate the site certificates when we need them. This will just take the domain name as a parameter that will be inserted into the cert template and then signed by the secret key generated above. We could also generate a seperate key for each site, or generate a single key for use with all sites. This technique is quite general.

[036] A SSL module is responsible for the handling of all data passed through on a secure connection.

[037] FIG. 9 shows a flow-chart of an example process performed by a Non-SSL Module. This module is responsible for the handling of all data passed through that is not

on a secure connection. Generally, its responsibility will be adding and removing headers as well as streaming the data through the filter module and the anonymizer proxy. During the processing of the request this module will prune the url down to its top level domain and enter it into the recently visited sites cache (Recently visited sites) It will also run the top level domain through the per-site settings cache to see if there is a match. If there is we will apply those security settings to this request if there isn't we will use the global settings (Security Level) . Once we have applied all of our security filters we will forward the request to the anonymizer proxy. When we receive the response from the anonymizer proxy we will do a second pass on our filters applying those that are relevant to incoming requests. Finally we will return the requested web resource to the browser.

[038] A Filtering Module is the module responsible for filtering the content requested by the user. If will remove items such as ads, popups etc... (Refer to the requirements features matrix). We will update the PagesBlocked, ActveContentBlcked, AdsBlocked and PopupsBlocked to reflect what we have done. This can be updated either on a per item (meaning every time something is blocked you update the count) or on a per page basis (meaning you keep counts for every page and update the counts after the page is finished loading). If while filtering we encounter a request that is on the blocked list we will redirect to Never_List_Redirect_URL if it is a url or redirect to Never_List_Redirect_Image if it is a request for an image. If OS hiding is active we will replace the OS in the headers with OS_Hiding_Name. Similarly if referrer hiding is active we will replace the referrer header with Referring_Hiding_Name.

[039] As per request preferences, the Client has to pass the following settings to the Proxy for each web-page request: IP-Hiding; SSL Fulltime; Active X Filter; Java Filter; JS filter; VB filter; Safe cookies; and other options or preferences. The client should send an HTTP header with each web request that will contain a bit-mask specifying what security options will be applied by the proxy on the returned web-page that the use has requested such as:

[040] For example in URL encryption, the the public key is stored by anonpro client in a cookie. The proxy will then pick up this cookie. (Currently the toolbar does it this way – but not with an XML structured cookie).

[041] Preferences

[042] Trusted List / Protected List

[043] - check "Blocked" list, then check "protected" list, then check "trusted" list

[044] - if URL is listed in the "blocked" list

[045] if it is the HTML mail page – popup a warning

[046] if it is just an image, replace the image with an "anonpro" image

[047] If the URL is listed in the "protected" list:

[048] Check the "per site" preferences

[049] If no "per site" preferences, then use what the current preferences are.

[050] FIG. 10 shows a signal flow diagram for an example process performed by the invention for recently visited sities. Whenever a request is made for a web page the address is placed inside our cache, we may schedule a thread that will run somewhat frequently to go through the cache using our url pruning algorithm to strip down to top

level domains removing duplicate entries and placing any new domains into the recent hash map.

[051] It is appreciated by those skilled in the art that the AnonPro Client may run on multiple platforms including Microsoft® Windows 98®, Windows NT®, Windows 2000®, Windows XP®, Apple operating systems such as OS 7, OS 8, OS 9, OS 10, Unix based operating system, Linix based operating systems or any other similar type of operating system used at present or in the future. The software may operating in any language. It is also appreciated that this system is secure. Therefore, To make communication as secure as desired for the client, the system has two levels of end user access security. In the AnonPro Client, there are the following security components: Application Layer Security and Network Security. Because of the nature of the system there should be established a trust relationship between the users and the products servers (Anonymizer Proxy, Anonymizer Web-Server) to grant access privilege and subsequent ability to protect the transactions. Various security means are used to assure recipients that message comes from a sender whose identity is validated and that its contents have not been tampered with during the transmission such as, for example, User-ID/Password Authentication and SSL. Every product session is fully protected by using encryption technology. We may use Secure Sockets Layer (SSL), which is widely adopted standard in industry, to encrypt data transmitted between a client and the <product> data center. 128-bit RC4 encryption mechanism in SSL is used to provide strong encryption. For the web access application, you have to have a Web Browser that supports 128-bit SSL. All the most popular Browsers support 128-bit SSL in recent versions.

[052] Username and password is used in authentication, but password is not passed in clear over the network even though it is protected by SSL link. Instead, in AnonPro Client, it is hashed, and he resulting "digest" is then sent over the network. When it arrives at the server site, server checks it to against the saved digest on the server database.

[053] The processes described in may be performed by hardware or software. If the process is performed by software, the software may reside in software memory (not shown) in the controller, memory, or an removable memory medium. The software in memory may include an ordered listing of executable instructions for implementing logical functions (i.e., "logic" that may be implement either in digital form such as digital circuitry or source code or in analog form such as analog circuitry or an analog source such an analog electrical, sound or video signal), may selectively be embodied in any computer-readable (or signal-bearing) medium for use by or in connection with an instruction execution system, apparatus, or device, such as a computer-based system, processor-containing system, or other system that may selectively fetch the instructions from the instruction execution system, apparatus, or device and execute the instructions. In the context of this document, a "computer-readable medium" and/or "signal-bearing medium" is any means that may contain, store, communicate, propagate, or transport the program for use by or in connection with the instruction execution system, apparatus, or device. The computer readable medium may selectively be, for example but not limited to, an electronic, magnetic, optical, electromagnetic, infrared, or semiconductor system, apparatus, device, or propagation medium. More specific examples "a non-exhaustive

list" of the computer-readable medium would include the following: an electrical connection "electronic" having one or more wires, a portable computer diskette (magnetic), a RAM (electronic), a read-only memory "ROM" (electronic), an erasable programmable read-only memory (EPROM or Flash memory) (electronic), an optical fiber (optical), and a portable compact disc read-only memory "CDROM" (optical). Note that the computer-readable medium may even be paper or another suitable medium upon which the program is printed, as the program can be electronically captured, via for instance optical scanning of the paper or other medium, then compiled, interpreted or otherwise processed in a suitable manner if necessary, and then stored in a computer memory.

[054] In general, the system described provides for Consensual Man in the Middle Attack by using used to rewrite pages, on the fly creation of site SSL certificates, CA cert to sign all SSL site certificates, CA cert is generated per user, CA cert is automatically installed in the browser and SSL page rewriting. Where the SSL page rewriting included the Client decrypting SSL pages to rewrite before re-encrypting and sending to proxy or end web site.

[055] The system also provides for the Client to Insert information into data stream from browser to Internet through any kind of Header or by inserting cookies. The cookies may include authentication / access rights information and preferences information and utilize XML and encryption.

[056] The system may also provides for a TCP level hook for privacy service that includes the Hook redirecting traffic to a local proxy on the user's machine, the Client proxy redirecting traffic to Anonymizer proxy and the TCP hook allows IP hiding.

[057] The system may also provides for a Full time SSL without URL prefixing.

[058] The system may also provides for making cookies session only and/or change cookie expiration date.

[059] The system may also provides for gathering and generation Privacy Statistics that include Per site privacy statistics, a Privacy Analyzer real time threat display, and automated site threat analysis and rating.

[060] The system may also provides for setting per site privacy settings that include white lists, black lists, detailed custom settings, "Show details" functionality, recommended site settings list that include automatically updated and downloaded settings, and hard coded Site settings that can't be changed by user have preset defaults and an exception list for some sites.

[061] The system may also provides for the Client to keep a list of alternate access names / IP addresses for accessing servers. The Client may tries all addresses one after another and/or each user gets a different set of access addresses.

[062] The system may also provides allows install on many computers while detect and prevent multiple simultaneous users.

[063] The system may also provides allows Client Javascript [script] rewriting.

[064] The system utilizes a novel GUI design to manage information.

[065] While various embodiments of the application have been described, it will be apparent to those of ordinary skill in the art that many more embodiments and implementations are possible that are within the scope of this invention. Accordingly, the invention is not to be restricted except in light of the attached claims and their equivalents. The foregoing description of an implementation has been presented for purposes of illustration and description. It is not exhaustive and does not limit the claimed inventions to the precise form disclosed. Modifications and variations are possible in light of the above description or may be acquired from practicing the invention. For example, the described implementation includes software but the invention may be implemented as a combination of hardware and software or in hardware alone. Note also that the implementation may vary between systems. The claims and their equivalents define the scope of the invention.

PTO/SB/16 (10-01)
Approved for use through10/31/2002. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

PROVISIONAL APPLICATION FOR PATENT COVER SHEET

This is a request for filing a PROVISIONAL APPLICATION FOR PATENT under 37 CFR 1.53(c).

Express Mail Label No.	EU245163300US

INVENTOR(S)

Given Name (first and middle [if any])	Family Name or Surname	Residence (City and either State or Foreign Country)
Lance M.	Cottrell	San Diego, California
James A	Reynolds	Carlsbad, California
Darya	Mazandarany	San Diego, California

☑ *Additional inventors are being named on the* 1 *separately numbered sheets attached hereto*

TITLE OF THE INVENTION (500 characters max)

A NETWORK PRIVACY SYSTEM

Direct all correspondence to: **CORRESPONDENCE ADDRESS**

☐ Customer Number ____ → *Place Customer Number Bar Code Label here*

OR *Type Customer Number here*

☑ Firm or Individual Name	Francisco A. Rubio-Campos				
Address	The Eclipse Group				
Address	26895 Aliso Creek Road, Suite B-104				
City	Aliso Viejo	State	CA	ZIP	92656
Country	USA	Telephone	949-448-9410	Fax	714-948-8903

ENCLOSED APPLICATION PARTS *(check all that apply)*

☑ Specification *Number of Pages* 19
☑ Drawing(s) *Number of Sheets* 10
☐ Application Data Sheet. See 37 CFR 1.76
☐ CD(s), Number
☐ Other (specify)

METHOD OF PAYMENT OF FILING FEES FOR THIS PROVISIONAL APPLICATION FOR PATENT

☐ Applicant claims small entity status. See 37 CFR 1.27.
☑ A check or money order is enclosed to cover the filing fees
☑ The Commissioner is hereby authorized to charge filing fees or credit any overpayment to Deposit Account Number: 502542
☐ Payment by credit card. Form PTO-2038 is attached.

FILING FEE AMOUNT ($): $80.00

The invention was made by an agency of the United States Government or under a contract with an agency of the United States Government.

☑ No.
☐ Yes, the name of the U.S. Government agency and the Government contract number are: _____

Respectfully submitted,

SIGNATURE _____ Date 06/25/2003

TYPED or PRINTED NAME Francisco A. Rubio-Campos

TELEPHONE (949) 448-9410

REGISTRATION NO. *(if appropriate)* 45,358
Docket Number: IF03004USV

USE ONLY FOR FILING A PROVISIONAL APPLICATION FOR PATENT

This collection of information is required by 37 CFR 1.51. The information is used by the public to file (and by the PTO to process) a provisional application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 8 hours to complete, including gathering, preparing, and submitting the complete provisional application to the PTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, Washington, D.C. 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Box Provisional Application, Assistant Commissioner for Patents, Washington, D.C. 20231.

PROVISIONAL APPLICATION COVER SHEET
Additional Page

PTO/SB/16 (02-01)
Approved for use through 10/31/2002. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Docket Number	IF03004USV

INVENTOR(S)/APPLICANT(S)

Given Name (first and middle [if any])	Family or Surname	Residence (City and either State or Foreign Country)
Peleus	Uhley	Alameda, California
Gene	Nelson	Spring Valley, California

Number 2 of 2

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

PTO/SB/17 (05-03)
Approved for use through 04/30/2003. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

FEE TRANSMITTAL for FY 2003

Effective 01/01/2003. Patent fees are subject to annual revision.

☑ Applicant claims small entity status. See 37 CFR 1.27

TOTAL AMOUNT OF PAYMENT ($) 80.00

Compl te if Known	
Application Number	Unknown
Filing Date	June 25, 2003
First Named Inventor	Lance M. Cottrell et al.
Examiner Name	Not applicable
Art Unit	Unassigned
Attorney Docket No.	IF03004USV

METHOD OF PAYMENT *(check all that apply)*

☐ Check ☑ Credit card ☐ Money Order ☐ Other ☐ None

☑ Deposit Account:

Deposit Account Number: 502542

Deposit Account Name: The Eclipse Group

The Director is authorized to: (check all that apply)

☑ Charge fee(s) indicated below ☑ Credit any overpayments

☐ Charge any additional fee(s) during the pendency of this application

☐ Charge fee(s) indicated below, except for the filing fee to the above-identified deposit account.

FEE CALCULATION

1. BASIC FILING FEE

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description	Fee Paid
1001	750	2001	375	Utility filing fee	
1002	330	2002	165	Design filing fee	
1003	520	2003	260	Plant filing fee	
1004	750	2004	375	Reissue filing fee	
1005	160	2005	80	Provisional filing fee	80.00

SUBTOTAL (1) ($) 80.00

2. EXTRA CLAIM FEES FOR UTILITY AND REISSUE

	Extra Claims		Fee from below		Fee Paid
Total Claims		-20** =		X	=
Independent Claims		- 3** =		X	=
Multiple Dependent					=

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description
1202	18	2202	9	Claims in excess of 20
1201	84	2201	42	Independent claims in excess of 3
1203	280	2203	140	Multiple dependent claim, if not paid
1204	84	2204	42	** Reissue independent claims over original patent
1205	18	2205	9	** Reissue claims in excess of 20 and over original patent

SUBTOTAL (2) ($) 0.00

**or number previously paid, if greater; For Reissues, see above

FEE CALCULATION (continued)

3. ADDITIONAL FEES

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description	Fee Paid
1051	130	2051	65	Surcharge - late filing fee or oath	
1052	50	2052	25	Surcharge - late provisional filing fee or cover sheet	
1053	130	1053	130	Non-English specification	
1812	2,520	1812	2,520	For filing a request for *ex parte* reexamination	
1804	920*	1804	920*	Requesting publication of SIR prior to Examiner action	
1805	1,840*	1805	1,840*	Requesting publication of SIR after Examiner action	
1251	110	2251	55	Extension for reply within first month	
1252	410	2252	205	Extension for reply within second month	
1253	930	2253	465	Extension for reply within third month	
1254	1,450	2254	725	Extension for reply within fourth month	
1255	1,970	2255	985	Extension for reply within fifth month	
1401	320	2401	160	Notice of Appeal	
1402	320	2402	160	Filing a brief in support of an appeal	
1403	280	2403	140	Request for oral hearing	
1451	1,510	1451	1,510	Petition to institute a public use proceeding	
1452	110	2452	55	Petition to revive - unavoidable	
1453	1,300	2453	650	Petition to revive - unintentional	
1501	1,300	2501	650	Utility issue fee (or reissue)	
1502	470	2502	235	Design issue fee	
1503	630	2503	315	Plant issue fee	
1460	130	1460	130	Petitions to the Commissioner	
1807	50	1807	50	Processing fee under 37 CFR 1.17(q)	
1806	180	1806	180	Submission of Information Disclosure Stmt	
8021	40	8021	40	Recording each patent assignment per property (times number of properties)	
1809	750	2809	375	Filing a submission after final rejection (37 CFR 1.129(a))	
1810	750	2810	375	For each additional invention to be examined (37 CFR 1.129(b))	
1801	750	2801	375	Request for Continued Examination (RCE)	
1802	900	1802	900	Request for expedited examination of a design application	

Other fee (specify) _____

*Reduced by Basic Filing Fee Paid

SUBTOTAL (3) ($) 0.00

SUBMITTED BY

				(Complete *if applicable*)	
Name (Print/Type)	Francisco A. Rubio-Campos	Registration No. (Attorney/Agent)	45,358	Telephone	949-448-9410
Signature				Date	June 25, 2003

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

Document made available under the Patent Cooperation Treaty (PCT)

International application number: PCT/US04/020562

International filing date: 25 June 2004 (25.06.2004)

Document type: Certified copy of priority document

Document details: Country/Office: US
Number: 60/482,784
Filing date: 25 June 2003 (25.06.2003)

Date of receipt at the International Bureau: 25 November 2004 (25.11.2004)

Remark: Priority document submitted or transmitted to the International Bureau in compliance with Rule 17.1(a) or (b)

World Intellectual Property Organization (WIPO) - Geneva, Switzerland
Organisation Mondiale de la Propriété Intellectuelle (OMPI) - Genève, Suisse